On the Influence of Description Logics Ontologies on Conceptual Similarity

Claudia d'Amato[1], Steffen Staab[2], and Nicola Fanizzi[1]

[1] Department of Computer Science, University of Bari, Italy
{claudia.damato,fanizzi}@di.uniba.it
[2] ISWeb, University of Koblenz-Landau, Germany
staab@uni-koblenz.de

Abstract. Similarity measures play a key role in the Semantic Web perspective. Indeed, most of the ontology related operations such as ontology learning, ontology alignment, ontology ranking and ontology population are grounded on the notion of similarity. In the last few years several similarity functions have been proposed for measuring both concept similarity and ontology similarity. However, they lack of a comprehensive formal characterization that is able to explain their behavior and value added, in particular when the ontologies are formulated in description logics languages like OWL-DL. Concept similarity functions need to be able to deal with the high expressive power of the ontology representation language, and to convey the underlying semantics of the ontology to which concepts refer. We propose a semantic similarity measure for complex Description Logics concept descriptions that elicits the underlying ontology semantics. Furthermore, we theorize a set of criteria that a measure has to satisfy in order to be compliant with a semantic expected behavior.

1 Introduction

The notion of *similarity* has been active, prominent and seminal in the areas of cognitive psychology [29, 16, 14], knowledge acquisition [25], data management and information organization [15, 22, 17] for a long time. In the last years, the importance of the similarity notion has been highlighted also in the Semantic Web (SW) context. Indeed, most of the ontology related operations such as ontology learning, ontology alignment, ontology ranking and ontology population are grounded on an idea of similarity. However, the definition of similarity or dissimilarity measures[3] in the SW context is a topic that has not been deeply investigated [4]. One of the main problems is the necessity to cope with the high expressive power of OWL[4] that is the standard ontology representation language. It is grounded on *Description Logics* (DLs) that are a family of logic languages characterized by a well defined formal semantics and a set of reasoning operators that are used for making explicit the knowledge that is implicitly asserted in the

[3] Since a dissimilarity measure can be always obtained from a similarity measure (see [3]) in the following we will consider only the notion of similarity measure

[4] www.w3.org/2004/OWL/

knowledge base (KB). Differently from the measures formalized in other contexts, similarity measures for ontological knowledge need to be able to deal with the semantics of the compared objects (concepts, individuals, ontologies).

In the last few years, several measures for assessing concept similarity and/or ontology similarity have been proposed. They are mainly an adaptation of measures defined for different and less expressive representations (such as feature vectors, trees, graphs (see [7] for the discussion)) to the SW context. As such, most of them are not able to convey the underlying semantics of the ontological representation. On the contrary, it is common wisdom that similarity measures for ontological knowledge need to be influenced by the reference ontology. However, this influence has never been put on a solid foundation with objective criteria.

Another important aspect, less considered in the literature, is the ability of a measure to assess similarity between the individuals of an ontology. Indeed, tasks such as clustering and ranking most of the times focus on the resources rather than on their conceptual descriptions. Furthermore, the purpose of a similarity measure is rarely the modeling of similarity on a scale. More often similarity is a means to an end, such as grouping the most similar entities for obtaining a more meaningful conceptual exploration or more efficient data management [11], ranking retrieved resources on the ground of their relevance w.r.t. a request [19] or mapping or aligning ontologies [13]. For these tasks, the relative comparison of two similarity values is more important than the absolute value of either. A similarity measure has to be able to characterize such relative comparisons in a way that accommodates to the given ontological definitions.

In this paper, we first of all discuss the intended behavior of a *semantic* (dis)similarity measure when it is applied to ontological knowledge (see Sect. 2). Then, we summarize the most widely used approaches for computing concept (dis)similarity and we show that, even if the measures defined in these approaches satisfy the mathematical definition of similarity function [3], they are not able to exploit the underlying ontology semantics, thus sometimes failing the correct assessment of the similarity value (see Sect. 3). To overcome this issue, we formalize a set of criteria that a measure has to satisfy and propose a measure for DL complex concept descriptions that is compliant with such formalization (see Sect. 5). Conclusions are drawn in Sect. 6.

2 Semantic Similarity Measures: Expected Behaviors

Ontologies represent a formal conceptualization of a certain domain where the meaning of the concepts is defined and the relationships among them are specified. The most important aspect of the ontological representation is its capacity of expressing domain semantics. Measures for estimating concept similarity have to be able to appropriately consider concept semantics in order to correctly assess their similarity value.

Let us consider the following example where the TBox[5] $\mathcal{T}$ and the ABox $\mathcal{A}$ concerning a few airport locations and air connections between them are specified:

$\mathcal{T} = \{\mathsf{Service} \sqsubset \mathsf{Top};\ \mathsf{Airport} \sqsubset \mathsf{Top} \sqcap \neg\mathsf{Service}; \mathsf{Town} \sqsubset \mathsf{Top} \sqcap \neg\mathsf{Service} \sqcap \neg\mathsf{Airport};$

[5] The TBox is a set of concept descriptions while the ABox contains the set of assertions concerning the world state, namely concepts and roles. For more details see the appendix A.

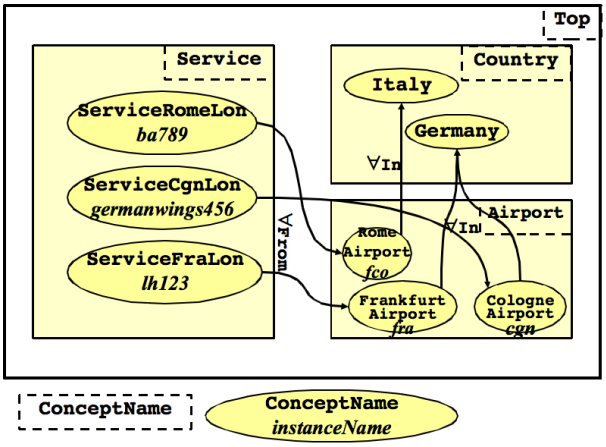

Fig. 1. Service and instance descriptions

$$\text{Country} \sqsubset \text{Top} \sqcap \neg\text{Service} \sqcap \neg\text{Town} \sqcap \neg\text{Airport};\ \text{Germany} \sqsubset \text{Country};$$
$$\text{Italy} \sqsubset \text{Country} \sqcap \neg\text{Germany};\ \text{UK} \sqsubset \text{Country} \sqcap \neg\text{Germany} \sqcap \neg\text{Italy};$$
$$\text{CologneAirport} \sqsubset \text{Airport} \sqcap \forall\text{In.Germany};\ \text{RomeAirport} \sqsubset \text{Airport} \sqcap \forall\text{In.Italy};$$
$$\text{FrankfurtAirport} \sqsubset \text{Airport} \sqcap \forall\text{In.Germany} \sqcap \neg\text{CologneAirport};$$
$$\text{LondonAirport} \sqsubset \text{Airport} \sqcap \forall\text{In.UK}\ \}$$

$$\mathcal{A} = \{\text{FrankfurtAirport(fra)}; \text{CologneAirport(cgn)}; \text{RomeAirport(fco)}; \text{LondonAirport(lhr)}\}$$

Based on this ontology, some new concepts and instances may be formulated as follows. The principle situation, abstracting from some details, is depicted in Figure 1.

$$\text{ServiceFraLon} = \text{Service} \sqcap \exists\text{From.FrankfurtAirport} \sqcap \forall\text{From.FrankfurtAirport} \sqcap$$
$$\sqcap \exists\text{To.LondonAirport} \sqcap \forall\text{To.LondonAirport}$$
$$\text{ServiceCgnLon} = \text{Service} \sqcap \exists\text{From.CologneAirport} \sqcap \forall\text{From.CologneAirport} \sqcap$$
$$\sqcap \exists\text{To.LondonAirport} \sqcap \forall\text{To.LondonAirport}$$
$$\text{ServiceRomeLon} = \text{Service} \sqcap \exists\text{From.RomeAirport} \sqcap \forall\text{From.RomeAirport} \sqcap$$
$$\sqcap \exists\text{To.LondonAirport} \sqcap \forall\text{To.LondonAirport}$$
$$\text{ServiceFraLon}(\textit{lh456});$$
$$\text{ServiceCgnLon}(\textit{germanwings123});$$
$$\text{ServiceRomeLon}(\textit{ba789})$$

Each concept description introduces a flight service respectively from Frankfurt, Cologne and Rome to London. For each concept, a corresponding instance is specified.

Expressive ontology languages, such as OWL, let us now express interesting questions like "which service (at the concept level) brings us to London?" or "which concrete service instantiation runs into London Heathrow Airport (lhr)?". Given an example like *lh456* at the instance level, an obvious question is, e.g. if *lh456* is booked out, which available service is more similar to *lh456*? Or similarly, given an example like ServiceFraLon at the concept level, if e.g. Frankfurt airport is not usable due to

the presence of snow, which available service is more similar to ServiceFraLon? Intuitively, at the instance level, *germanwings123* should be favored over *ba789* and, at concept level, ServiceCgnLon should be favored over ServiceRomeLon, since from the KB, we know that FrankfurtAirport and CologneAirport are both Airports in Germany, which is disjoint from Italy, where RomeAirport is located. A similarity measure has to be able to catch such a similarity. In order to do this, it needs to appreciate the underlying ontology semantics (that allows to know that FrankfurtAirport and CologneAirport are both German airports). We will call this expected behavior of a similarity measure *soundness*.

Now, let us assume to add the following criteria to the KB presented above:

$\{\mathsf{ItalianAirport} \sqsubseteq \mathsf{Airport};\ \mathsf{RomeAirport} \sqsubseteq \mathsf{ItalianAirport};\ \mathsf{GermanAirport} \sqsubseteq \mathsf{Airport};\ \mathsf{FrankfurtAirport} \sqsubseteq \mathsf{GermanAirport};\ \mathsf{CologneAirport} \sqsubseteq \mathsf{GermanAirport}\ \}$

and the following concept definition:

$$\mathsf{ServiceItLon} = \mathsf{Service} \sqcap \exists \mathsf{From.RomeAirport} \sqcap \forall \mathsf{From.RomeAirport} \sqcap \\ \sqcap \forall \mathsf{From.ItalianAirport} \sqcap \exists \mathsf{To.LondonAirport} \sqcap \forall \mathsf{To.LondonAirport}$$

ServiceItLon is semantically equivalent (by applying the rewriting rules [12]) to ServiceRomeLon. Hence, the similarity value of each of them w.r.t. another concept should be equal, i.e. the similarity value of ServiceItLon and ServiceCgnLon should be equal to the similarity value of ServiceRomeLon and ServiceCgnLon. We will call this expected behavior of a similarity measure *equivalence soundness*.

Furthermore, similarity between disjoint concepts needs not always to be zero. Let us suppose to have the following axiom $\mathsf{ServiceCgnLon} \equiv \neg \mathsf{ServiceFraLon}$ in the KB. By comparing ServiceCgnLon and ServiceFraLon it can be noted that they are not totally different since both perform a flight from a German airport to London. Consequently, *sim*(ServiceCgnLon,ServiceFraLon) should be greater than, for instance, *sim*(ServiceCgnLon,Service) where the only thing that we know from the KB is that ServiceCgnLon is a Service. However, in the same way we know that ServiceRomeLon is a Service. This means that not so much semantic information is shared by ServiceCgnLon and Service, differently from ServiceFraLon and ServiceCgnLon that describe mainly the same service, the only difference is given by the starting airport. We will call the ability of a similarity measure to recognize similarities between disjoint concepts *disjointness incompatibility*.

In the following we show how most of the measures defined in the literature fail these expected behaviors. In order to avoid this phenomenon, we approach a formalization of a set of criteria that a similarity or a dissimilarity measure needs to satisfy.

3 Computing Concept Similarity: Related Works

Similarity and dissimilarity measures have been largely studied in the literature, considering several kinds of representations such as feature vectors, trees, graphs etc. Though specific (dis-)similiarity measures have been proposed for description logics, the question "*what constitutes a good semantic similarity measure in a description logics language*" has not been investigated so far. In the following, after recalling the formal

definition of (dis)similarity measure [3], the main approaches for computing concept similarity are analyzed: the extensional-based and the intentional-based approaches. Hence, the most representative measures for assessing concept similarity are considered. We show that, even if such measures satisfy the formal definition, they fail the expected behaviors of a *semantic* similarity measure (see Sect. 2).

3.1 Similarity Measure: Mathematical Foundation

Similarity and dissimilarity measures are applied to objects of a considered domain for determining how much similar and different respectively they are. Intuitively, defining a similarity measure requires two steps. In the first one, a set of similarity values, e.g. $\{$*far, near*$\}$ or $\{$*equal, similar, dissimilar, totally different*$\}$, is defined. A commonly used scale is the set of the real number $\mathbb{R}$. The second step consists in defining a function from a pair of objects to the set of similarity values. Following [3], formal definitions of similarity and dissimilarity measure can be given.

Definition 1 (Similarity Measure). *Let D be a set of elements of a considered domain and let $(V, \leq)$ be a totally ordered set. A function $s : D \times D \rightarrow V$ is a similarity function iff there exists an element $0_V \in V$ and an element $1_V \in V$ such that:*

1. $\forall x, y \in D \ : \ s(x, y) \geq 0_V$ *(positiveness)*
2. $\forall x \in D \ : \ s(x, x) = 1_V$ *and* $\forall y \in D \wedge x \neq y \ : \ s(x, x) \geq s(x, y)$ *(reflexivity)*
3. $\forall x, y \in D \ : \ s(x, y) = s(y, x)$ *(symmetry)*

Definition 2 (Dissimilarity Measure). *Let D be a set of elements of a considered domain and let $(V, \leq)$ be a totally ordered set. A function $d : D \times D \rightarrow V$ is a dissimilarity function iff there exists an element $0_V \in V$ such that:*

1. $\forall x, y \in D \ : \ d(x, y) \geq 0_V$ *(positiveness)*
2. $\forall x \in D \ : \ d(x, x) = 0_V$ *and* $\forall y \in D \wedge x \neq y \ : \ d(x, x) \leq d(x, y)$ *(reflexivity)*
3. $\forall x, y \in D \ : \ d(x, y) = d(y, x)$ *(symmetry)*

In the sequel the terms *measure* and *function* will be used interchangeably. For the next properties, only the dissimilarity measure will be considered. They can be easily obtained for the case of a similarity measure. It will be denoted with s a similarity measure and with d a dissimilarity measure.

Definition 3 (Strictness property). *Let D be a set of elements of a considered domain and let $(V, \leq)$ be a totally ordered set. Let d be a dissimilarity function on D with minimum value $0_V \in V$. The function d is strict iff $\forall x, y \in D \ : \ d(x, y) = 0_v \Rightarrow x = y$*

The strictness property ensures that the minimum dissimilarity value is assigned only if the considered elements are equal. If the strictness property is not satisfied then also elements that are different could assume the lowest dissimilarity value.

Definition 4 (Triangle inequality property). *Let D be a set of elements of a considered domain and let $(V, +, \leq)$ be a totally ordered set equipped with an order-preserving addition operation s.t. $(V, +)$ is a commutative group. Let d be a dissimilarity function on D. The function d satisfies the triangle inequality iff $\forall x, y, z \in D \ : \ d(x, y) + d(y, z) \geq d(x, z)$*

Definition 5 (Pseudo-metric). *A dissimilarity function is a* pseudo-metric *(or equivalently a* semi-distance*) iff it satisfies the triangle inequality.*

Definition 6 (Metric). *A pseudo-metric is a* metric *(or equivalently a* distance*) iff it satisfies the strictness property.*

Definition 7 (Normalized Dissimilarity Function). *Let D be a set of elements of a considered domain. Let $d : D \times D \rightarrow \mathbb{R}$ be a dissimilarity function. d is a normalized dissimilarity function if $\forall x, y \in D \ : \ 0 \leq d(x, y) \leq 1$ where $d(x, x) = 0$ and $\forall x \in D, \exists y \in D \ : \ d(x, y) = 1$*

Given a dissimilarity function d, it is always possible to define the normalized dissimilarity function d'.

3.2 Extensional-based Similarity Measures

The extensional-based similarity measures are basically inspired from the Jaccard similarity measure [18] and the Tversky's *contrast model* [29]. Extensional measures mainly assign to the compared concepts a similarity value that is proportional to the overlap of the concept extensions, namely the sets of individuals that are instances of the considered concepts (see App. A for more details). In [9], a similarity measure for DL concept descriptions is proposed. The similarity value is computed as the ratio between the overlap of the concept extensions and their union, weighted with a factor representing how far the two concepts are from the equivalence or from the subsumption[6]. This measure (as all measures based on the overlap of concept extensions) fails the *soundness* criterion (see Sect. 2), namely it is not able to fully convey the underlying ontology semantics. Indeed, by considering the concepts ServiceFraLon and ServiceCgnLon from the example above, their similarity will be zero, since they do not share any instance. Instead, as observed in Sect. 2, we know from the KB that they are semantically more similar than other pairs, since they both fly from Germany to London, differently from a pair of flights that start in different countries.

Also similarity measures based on the notion of *Information Content* (*IC*) ultimately exploit the extension overlap. The first measures grounded on this notion have been proposed by Resnik [27, 28]. The main idea consists in measuring the similarity of concepts (represented in a is-a taxonomy) on the ground of the amount of information that they share. This is approximated with the quantity of information conveyed by the *most specific ancestor*[7] of the considered concepts that is measured by recurring to the notion of *IC*. The *IC* of a concept C is defined as: $IC(C) = -log\ p(C)$ where $p(C)$ is the probability of the concept C and it is computed by Resnik as the probability of occurrence of C in a corpus. The main issue of using *IC* for measuring the similarity of concepts in an ontology is how to compute the concept probability. In [10], it is approximated to the ratio of the concept extension and the extension of the entire ABox and it is used for computing the dissimilarity of $\mathcal{ALC}$ concept descriptions. Specifically, the dissimilarity function assigns the maximum value, that is 1, if the considered concepts are

[6] For the notion of concept equivalence and subsumption see App. A.

[7] The most specific ancestor is the first parent node of the considered concepts in the hierarchy.

disjoint; the minimum value, that is 0 if the two concepts are equivalent; otherwise it is recursively defined and the base step consists in measuring the dissimilarity between primitive concepts that is computed as the variation of the *IC* of the considered primitive concepts w.r.t. the *IC* of their *Least Common Subsumer* (LCS) (see Sect. A). This measure fails the *disjointness incompatibility* criterion (see Sect. 2), namely it is not able to recognize similarities between disjoint concepts. It is important to note that almost all the extensional-based similarity measures do not satisfy the disjointness incompatibility criterion. Indeed, since they are based on the overlapping of concept extensions, such an intersection will be always empty when disjoint concepts are considered.

3.3 Intentional-based Similarity Measures

The intentional-based similarity measures are functions that exploit the structure of the concept definitions for assessing their similarity. One of the most well known measure based on this approach has been proposed by Rada et al. [26]. It is based on the notion of path distance. Concepts are nodes linked by is-a edges[8] in a semantic network [6] having a tree structure. The similarity of two concepts C and D is computed as the length of the shortest path connecting C and D. Formally, $sim(C, D) = length(C, E) + length(D, E)$ where E is the *most specific ancestor*[9] (*msa* for brevity) of C and D and $length(C, E)$ is the number of edges that link the concepts C and E. The main drawback of this measure is that it is not able to cope with relationships that are more expressive than is-a (as typically occurs in OWL). Moreover, it is highly sensitive to the predefined hierarchical network; it tends to give coarse similarity values to concepts that have the same ancestor but it is not able to rate them. Let us consider the concepts ServiceFraLon, ServiceCgnLon and ServiceRomeLon from the example above, and their *msa* that is Service. The similarity between ServiceFraLon, ServiceCgnLon and the similarity between ServiceFraLon, ServiceRomeLon will be the same. This result violates the *soundness* criterion (see Sect. 2). Indeed, we know from the KB that ServiceFraLon and ServiceCgnLon are more semantically similar than ServiceFraLon and ServiceRomeLon because the former perform flights from a German airport to London, the latter perform flights starting from different countries.

A similar approach is used in [23], where the taxonomic overlapping between two hierarchical ontologies is computed. The notion of *semantic cotopy* (SC) of a concept C is introduced; it is given by the set of all direct super and sub-concepts of C in the ontology $\mathcal{O}$ where C is defined. Given the SC of C in $\mathcal{O}_1$ and the SC of C in $\mathcal{O}_2$, the taxonomic overlapping[10] of $\mathcal{O}_1$ and $\mathcal{O}_2$ w.r.t. C is computed as the ratio between the intersection of the two SCs and their union. As in the previous case, this measure strongly depends from the predefined taxonomy, thus it can fail the *soundness* criterion.

Other intentional-based similarity measures compute concept similarity by comparing the syntactic concept descriptions. In [8], a dissimilarity measure for $\mathcal{ALC}$ concept

[8] Based on the same rationale, an extension of such a measure, that is able to consider other kinds of relationships, has been also proposed.

[9] Note that, since the considered semantic network is a tree, the *msa* of two given nodes always exists and it is unique.

[10] The overall taxonomic overlapping between $\mathcal{O}_1$ and $\mathcal{O}_2$ is given by the averaged taxonomic overlapping computed w.r.t. to all concepts in the ontologies.

descriptions is presented. Concepts are assumed to be in $\mathcal{ALC}$ normal form [5], namely they are expressed as disjunction of conjunctive concepts, where the conjunctive concepts can be primitive, universal and existential concept restrictions. Given two concepts C and D, the measure returns the maximum dissimilarity value computed between all possible combination of disjunctive elements in C and D, namely the dissimilarity value computed at the conjunctive level. The dissimilarity value at conjunctive level is given by the sum of the dissimilarity values computed between: universal concept restrictions, existential concept restrictions and primitive concepts. The measure is recursively defined; the ground step is given by the computation of the dissimilarity between primitive concepts that is inversely proportional to the amount of extension overlap. On the ground of this function, similarity measures for $\mathcal{ALCNR}$ and $\mathcal{ALCHQ}$ normal form concept descriptions have been proposed [20, 21]. Measures based on the syntactic comparison of concept definitions fail the *equivalence soundness* (see Sect. 2). Indeed, given the concept definition Parent $\equiv$ Human $\sqcap$ $\exists$hasChild.Human and the concept descriptions Parent$\sqcap$Man and Human $\sqcap$ $\exists$hasChild.Human $\sqcap$ Man it is straightforward to see that these concept descriptions are equivalent. However, by measuring the similarity of each of them w.r.t. a third concept description i.e. Parent $\sqcap$ Man $\sqcap$ $\exists$hasChild.(Human $\sqcap$ $\neg$ Man), we will find a different similarity value since they are written in two different ways.

In [19] a dissimilarity measure for $\mathcal{SHIF}$ and $\mathcal{SHOIN}$ concept descriptions is presented. Given two concepts C and D, they are unfolded so that only primitive concept and role names appear in the definition. Hence, each concept is described by means of a feature vector where each feature is a primitive concept name or a primitive role name and its value is given by the (weighted) number of occurrences in the unfolded concept description. Once that the feature vectors are obtained, concept dissimilarities are computed as vector distances in high dimensional space. The role of the unfolding is to make explicit the concept semantics. This measure fails the *soundness* criterion since it is not able to fully exploit the knowledge in the reference ontology. Indeed, by considering the concepts ServiceFraLon and ServiceCgnLon from the example above, the unfolding does not take advantage of the fact that CologneAirport and FrankfurtAirport are German airports since inclusion criteria are only used.

Tab. 1 summarizes the similarity measures recalled in this section and their behavior w.r.t. the semantic criteria presented in Sect. 2.

4 Semantic Measure: Characterization

In this section, we define the formal criteria of *equivalence soundness* and *(strict) monotonicity* that allow us to deal with the issues raised in Sect. 2, namely *soundness*, *equivalence soundness*, *disjointness incompatibility*. Specifically, we define a set of criteria that a similarity measure[11] has to satisfy in order to be compliant with the expected behavior presented in Sect. 2. The criteria are specified considering the case of a dissimilarity measure. They are almost the same if a similarity measure is considered.

[11] Note that a similarity measure is considered. Hence, the criteria of the formal definition (see Def. 1 and Def. 2) have to be satisfied.

Table 1. Intentional and extensional based similarity measures and their behavior w.r.t. semantic criteria. "$\surd$" stands for criterion satisfied; "X" stands for criterion not satisfied.

	MEASURE	*Soundness*	*Equiv. soundness*	*Disj. Incompatibility*
EXT.	d'Amato et al. [9]	$\surd$	X	$\surd$
	d'Amato et al. [10]	X	X	$\surd$
INT.-BASED	Rada et al. [26]	$\surd$	X	X
	Maedche et al. [23]	$\surd$	X	X
	d'Amato et al. [8]	X	$\surd$	$\surd$
	Janowicz et al. [20, 21]	X	$\surd$	X
	Hu et al. [19]	$\surd$	X	X

Criterion 1 (Equivalence Soundness) *Let $(\mathcal{C}, d)$ a metric space where $\mathcal{C}$ is the set of DL concept descriptions expressible in the given language. A dissimilarity measure $d : \mathcal{C} \times \mathcal{C} \rightarrow [0, 1]$ obeys the criterion of equivalence soundness iff:* $\forall C, D, E \in \mathcal{C} : D \equiv E \Rightarrow d(C, D) = d(C, E)$.

This criterion simply requires that if two concepts are equivalent than the dissimilarity value of each of them w.r.t. a third concept has to be equal. This means, for instance, that, given the concept Father defined as Parent $\sqcap$ Man and the concept Child then *dis*(Father,Child) = *dis*(Parent $\sqcap$ Man,Child). Even if this is a quite obvious behavior, as seen in Sect. 3, most of the existing measures do not satisfy this criterion. In the following a proposition for helping the proof of the *equivalence soundness* is reported:

Proposition 1. *If the triangle inequality holds for a given dissimilarity measure d then it satisfies the* equivalence soundness *axiom.*

Proof. Given the concepts $C, D, E \in \mathcal{C}$ and a dissimilarity measure d, as the triangle inequality holds, it is true that: (1) $d(C, D)+d(D, E) \geq d(C, E)$ and also (2) $d(C, E)+ d(E, D) \geq d(C, D)$. If $D \equiv E$ then, according to Def. 2 it holds that $d(D, E) = 0 = d(E, D)$. Hence, $d(C, D) \geq d(C, E) \geq d(C, D) \Rightarrow d(C, D) = d(C, E)$.

The proposition is analogously proved if a similarity measure s is considered.

In the following, the *monotonicity* criterion is presented. It formalizes the monotonic behavior of the measure w.r.t. the specificity/generality of the considered concepts in the KB. We show how the *monotonicity* criterion generalizes the notion of *soundness* and *disjointess incompatibility* presented in Sect. 2.

Criterion 2 (Monotonicity) *Let $(\mathcal{C}, d)$ a metric space, $\mathcal{C}$ is the set of DL concept descriptions in the given language. A dissimilarity measure $d : \mathcal{C} \times \mathcal{C} \rightarrow [0, 1]$ obeys the monotonicity criterion iff given the concept expressions $C, D, E, L, U \in \mathcal{C}$ s.t:*

1. $C \sqsubseteq L, D \sqsubseteq L, C \sqsubseteq U, D \sqsubseteq U$,
2. $E \sqsubseteq U$, *and* $E \not\sqsubseteq L$
3. $\nexists H \in \mathcal{C}$ *s.t.* $C \sqsubseteq H \wedge E \sqsubseteq H \wedge D \not\sqsubseteq H$

imply that $d(C, D) \leq d(C, E)$.

This criterion asserts that, if given the concepts C, D and E, the concept generalizing C and D is more specific (w.r.t. the subsumption relationship) than the concept generalizing C and E, than C and D are more similar to each other w.r.t. C and E or equivalently C and D have a lower dissimilarity value w.r.t. C and E.

This criterion of *monotonicity* covers the notion of *soundness* introduced in Sect. 2 which requires the ability of a measure to convey the underlying ontology semantics. Indeed, by considering the concepts ServiceCgnLon, ServiceFraLon and ServiceRomeLon defined in Sect. 2, due to Criterion 2 we have that *dis*(ServiceCgnLon, ServiceFraLon) should be lower than *dis*(ServiceCgnLon, ServiceRomeLon) since there exists the concept Service $\sqcap \exists$ From.(Airport $\sqcap \forall$ In.Germany) $\sqcap \forall$ From.(Airport $\sqcap \forall$ In.Germany) $\sqcap \exists$ To.LondonAirport $\sqcap \forall$ To.LondonAirport generalizing ServiceCgnLon and ServiceFraLon that is more specific than Service generalizing ServiceCgnLon and ServiceRomeLon. This result is coherent with the semantic information conveyed from the KB, from which we know that ServiceCgnLon and ServiceFraLon are more similar than ServiceCgnLon and ServiceRomeLon since both ServiceCgnLon and ServiceFraLon describe a flight from a German airport to London, differently from ServiceCgnLon and ServiceRomeLon that describe a flight starting from two different countries.

Moreover, the *monotonicity* criterion also captures the notion of *disjointness incompatibility* which requires that if two concepts are disjoint their similarity is not necessarily null (or equivalently their dissimilarity is not necessarily maximal). This is straightforwardly verified by noting that, if the following disjointness axiom ServiceCgnLon $\equiv \neg$ ServiceFraLon is considered, the relation *dis*(ServiceCgnLon, ServiceFraLon) $\leq$ *dis*(ServiceCgnLon, ServiceRomeLon) remains valid. This is further evident by considering the following criterion:

Criterion 3 (Strict Monotonicity) *Let $(\mathcal{C}, d)$ a metric space where $\mathcal{C}$ is the set of DL concept descriptions expressible in the given language. A dissimilarity measure $d : \mathcal{C} \times \mathcal{C} \rightarrow [0, 1]$ obeys the soundness and disjointness incompatibility criteria iff given the concept expressions $C, D, E, L, U \in \mathcal{C}$ s.t:*

1. $C \sqsubset L, D \sqsubset L, C \sqsubset U, D \sqsubset U$,
2. $E \sqsubset U$, *and* $E \not\sqsubset L$
3. $\nexists H \in \mathcal{C}$ *s.t.* $C \sqsubset H \wedge E \sqsubset H \wedge D \not\sqsubset H$

imply that $d(C, D) < d(C, E)$.

Considering the concepts ServiceCgnLon, ServiceFraLon, ServiceRomeLon, due to Crit. 3, the relation *dis*(ServiceCgnLon, ServiceFraLon) $<$ *dis*(ServiceCgnLon, ServiceRomeLon) is valid although ServiceCgnLon and ServiceFraLon do not have common instances. Therefore *Strict Monotonicity* criterion allows that also non-empty extension intersections may still lead to a dissimilarity lower than one, that is the maximum dissimilarity value.

Criteria 2 and 3 pose an open issue: "*how to compute a concept generalization that is able to take into account both the concept definitions and the TBox of reference?*". A first reply could be to consider the Least Common Subsumer (LCS) (see App. A) of the considered concepts. Anyway, this is not the right solution. Indeed, for DLs

allowing for concept disjunction, the LCS is given by the disjunction of the considered concepts. It is not the suitable generalization because: 1) it does not take into account the TBox of reference; 2) it does not add further information besides of the information given by the considered concepts. If less expressive DLs (i.e. those do not allow for concept disjunction) are considered, again the LCS does not take into account the TBox of reference and moreover it is computed in a structural way, namely by considering the common concept and role names that appear in the concept definitions. A possible generalization able to satisfy our requirements is the Good Common Subsumer (GCS) (see App. A). Anyway, it is defined only for $\mathcal{ALE}(\mathcal{T})$ concept descriptions. If most expressive DLs are considered the problem remains still open.

5 The GCS-based Semantic Similarity Measure

From the discussion about the expected behaviors of a semantic similarity measure (Sect. 2) and the violation of such behaviors of the extensional-based and the intentional-based similarity measures, a possible conclusion is that a *semantic similarity measure* should be defined in a way that is neither structural nor extensional. Moving from this intuition we propose a semantic similarity measure that exploits the notion of concept extension, but instead of counting the common instances between two considered concepts, it assesses the similarity value as the variation of the number of instances in the concept extensions w.r.t. the number of instances in the extension of their common super-concept. The common super-concept is given by the GCS of the considered concepts (see Sect. A). The new similarity measure, the **GCS-based similarity**, is able to satisfy the semantic criteria presented in Sec. 4 and to exploit the underlying ontology semantics. The measure is formally defined in the following [11].

Definition 8 (GCS-based Similarity Measure). *Let $\mathcal{T}$ be an $\mathcal{ALC}$ TBox. For all C and D $\mathcal{ALE}(\mathcal{T})$-concept descriptions, the* Semantic Similarity Measure *s is a function $s : \mathcal{ALE}(\mathcal{T}) \times \mathcal{ALE}(\mathcal{T}) \rightarrow [0, 1]$ defined as follows:*

$$s(C, D) = \frac{\min(|C^I|, |D^I|)}{|(GCS(C, D))^I|} \cdot (1 - \frac{|(GCS(C, D))^I|}{|\Delta^I|} \cdot (1 - \frac{\min(|C^I|, |D^I|)}{|(GCS(C, D))^I|}))$$

where $(\cdot)^I$ computes the concept extension w.r.t. the interpretation I.

As interpretation, the canonical interpretation is considered that adopts the set of individuals in the ABox as domain and the identity as interpretation function [12, 24].

The rationale of the measure is that if two concepts are semantically similar, such as credit-card-payment and debit-card-payment, then they should have a good common superconcept, e.g. card-payment, that is close to the two concepts, namely the extensions of the superconcept and even the lesser-sized input concept share many instances. Consequently the similarity value will be close to 1. On the contrary, if the concepts are very different, e.g. car-transfer and debit-card-payment, their GCS, e.g. service, will be high up in the TBox, and it will have many instances that are not contained in the two compared concepts. Consequently the similarity value will be next to 0. In Fig. 2 and Fig. 3 this rationale is illustrated.

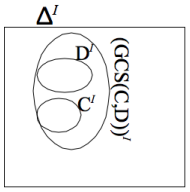

Fig. 2. Concepts C ≡credit-card-payment, D ≡debit-card-payment are similar as the extension of their GCS≡card-payment does not include many other instances besides of those of C and D.

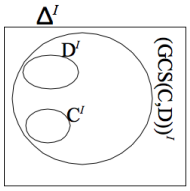

Fig. 3. Concepts C ≡car-transfer, $D \equiv$ debit-card-payment are different as the extension of their GCS≡service includes many other instances besides of those of C and D extensions.

The minimum concept extension is considered in the definition, in order to avoid incorrect similarity values (high similarity value) when one of the concepts, let say C, is very similar to the super-concept but very different from the other one, let say D.

Since the GCS is used in the measure definition, only $\mathcal{ALC}$ and $\mathcal{ALE}(\mathcal{T})$ DLs have been considered. If a different way for determining the common super-concept is adopted, other DLs can be used. The GCS has been chosen since it is not so specific as the $\mathcal{ALC}$ LCS, that is simply given by the disjunction of the concepts, neither it is too much general as the $\mathcal{ALE}$ LCS computed without taking into account the KB.

Differently from the existing similarity measures, the function defined above does not require the extension overlap of the compared concepts, and it does not take into account neither the path distance nor the structural comparison of concept defintions.

The GCS-based function is really a similarity measure: it is a positive definite function (from the definition), it is symmetric (this is ensured by the commutativity of the operators used in the definition) and it assumes the maximum value, which is 1, when the compared concepts are equivalent, indeed, only in this case the GCS will be equivalent to the considered concepts, and consequently, the extensions will be all equal.

Moreover, the GCS-based similarity measure is a semantic similarity measure. Indeed, given two equivalent concepts D and E and a third concept C, for the GCS definition [2], the GCS of C and D will be equivalent to the GCS of C and E and this ensures that the *equivalence soundness* criterion is satisfied. In the same way, the *disjointness incompatibility* criterion is also satisfied. Let us assume the following TBox $\mathcal{T} = \{$Human⊑ Top; Female⊑ Top; Male⊑ Top; Table⊑ Top; Woman≡ Human ⊓ Female; Man≡ Human ⊓ Male;$\}$ and the concepts Woman and Man (that are declared to be disjoint in the KB) having Human as GCS. By applying the GCS-based measure to the concepts Woman and Man we find that their similarity value is not null, which satisfies the disjointness incompatibility criterion. The *soundness* criterion is straightforwardly satisfied by considering the GCS as concept generalization. Indeed, by measuring the similarity between the concepts ServiceFraLon and ServiceCgnLon (defined in Sect. 2) and the similarity between ServiceCgnLon and Service we find that the former couple has a higher similarity value w.r.t. the latter. Futhermore, since the GCS is used in the concept definition, this allows to "understand" that ServiceCgnLon and ServiceFraLon are both flight performed from a German airport.

The GCS-based similarity can be also used for measuring individual similarity by first computing the *Most Specific Concepts* (see App. A) of the individuals to compare.

6 Conclusions

In this paper we have analyzed the attended behaviors that a similarity measure should have when it is used for measuring concept similarity in ontological knowledge. We have formalized such behaviors by introducing the notions of *(equivalence) soundness* and *disjointness incompatibility*. Hence we have showed that most of the measures currently used do not fully satisfy these notions. For this reason, we have formalized a set of criteria (*equivalence soundness*, *(strict) monotonicity*) that a measure needs to fulfil in order to be compliant with the attended behavior and we have introduced a new semantic similarity measure satisfying these criteria. This measure is based on the notion of Good Common Subsumer and it is able to exploit the semantics of the underlying ontology to which concepts refer. The measure is grounded on the concept extensions but, differently from the current approaches, it does not evaluate the extension overlap but the variation of the cardinality of the extensions of the considered concepts w.r.t. the cardinality of the extension of their GCS. This allow to overcome the limitations of the extensional-based similarity measures, while the use of the GCS allows to overcome the drawback of the intentional-based measures.

Acknowledgments

This research was partially supported by the the regional interest projects DIPIS (*Distributed Production as Innovative System*) and by the European Commission under contract IST-2006-027595, Lifecycle Support for Networked Ontologies - NeOn. The expressed content is the view of the authors but not necessarily the view of the mentioned projects.

References

1. F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In T. Dean, editor, *Proc. of IJCAI*, pages 96–101. Morgan Kaufmann, 1999.
2. F. Baader, R. Sertkaya, and Y. Turhan. Computing least common subsumers w.r.t. a background terminology. In *Proc. of the Int. Workshop on DLs*. CEUR-WS.org, 2004.
3. H.H. Bock and E. Diday. *Analysis of symbolic data : exploratory methods for extracting statistical information from complex data.* Springer-Verlag, 2000.
4. A. Borgida, T.J. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In *Proc. of the Int. Description Logics WS*, volume 147 of *CEUR*, 2005.
5. S. Brandt, R. Küsters, and A.-Y. Turhan. Approximation and difference in description logics. In D. Fensel, F. Giunchiglia, D. McGuinness, and M.-A. Williams, editors, *Proc. of the Int. Conf. on Knowledge Representation*, pages 203–214. Morgan Kaufmann, 2002.
6. A. Collins and M. Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior.*, 8:240–247, 1969.
7. C. d'Amato. Similarity-based Learning Methods for the Semantic Web, 2007. PhD Thesis.

8. C. d'Amato, N. Fanizzi, and F. Esposito. A dissimilarity measure for concept descriptions in expressive ontology languages. In H. Alani, C. Brewster, N. Noy, and D. Sleeman, editors, *Proceedings of the KCAP2005 Ontology Management Workshop*, Banff, Canada, 2005.
9. C. d'Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. In A. Pettorossi, editor, *Proceedings of Convegno Italiano di Logica Computazionale (CILC05)*, Rome, Italy, 2005.
10. C. d'Amato, N. Fanizzi, and F. Esposito. A dissimilarity measure for $\mathcal{ALC}$ concept descriptions. In *Proceedings of the 21st Annual ACM Symposium of Applied Computing, SAC2006*, volume 2, pages 1695–1699, Dijon, France, 2006. ACM.
11. C. d'Amato, S. Staab, N. Fanizzi, and F. Esposito. Efficient discovery of services specified in description logics languages. In *Proc. of the ISWC Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web.*, 2007.
12. F. Baader et al., editor. *The Description Logic Handbook*. Cambridge University Press, 2003.
13. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in owl-lite. In R. L. de Mántaras and L. Saitta, editors, *Proc. of the 16th Eureopean Conference on Artificial Intelligence, ECAI2004*, pages 333–337. IOS Press, 2004.
14. R. L. Goldstone, D. L. Medin, and J. Halberstadt. Similarity in context. *Memory and Cognition*, 25(2):237–255, 1997.
15. S. Haykin. *Self-organizing maps: Neural networks - A comprehensive foundation 2nd Edition*. Prentice-Hall, 1999.
16. K. J. Holyoak and P. Thagard. *Mental leaps analogy in creative thought*. Cambridge, MA: MIT Press, 1995.
17. B. Hu, Y. Kalfoglou, H. Alani, D. Dupplaw, P. Lewis, and N. Shadbolt. Semantic metrics. In *Proc. of the Int. Conf. on Managing Knowledge in a World of Networks, EKAW 2006*, volume 4248 of *LNCS*, pages 166–181. Springer, 2006.
18. J. Hunter, J. Drennan, and S. Little. Realizing the hydrogen economy through semantic web technologies. *IEEE Intelligent Systems*, 19(1):40–47, 2004.
19. P. Jaccard. étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.
20. K. Janowicz. Sim-dl: Towards a semantic similarity measurement theory for the description logic $\mathcal{ALCNR}$ in geographic information retrieval. In R. Meersman et al., editor, *Proceedings of SeBGIS 2006, OTM Workshops*, volume 4278 of *LNCS*, pages 1681–1692, 2006.
21. K. Janowicz and C. Keßler and M. Schwarz and M. Wilkes and I. Panov and M. Espeter and B. Bäumer. Algorithm, Implementation and Application of the SIM-DL Similarity Server. In *Proc. of 2nd Int. Conf. on GeoSpatial Semantics (GeoS 2007)*, LNCS. Springer, 2007.
22. T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:881–892, 2002.
23. A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proc. of the Int. EKAW Conf.*, volume 2473 of *LNCS*, pages 251–263. Springer, 2002.
24. T. Mantay. Commonality-based ABox retrieval. Technical Report FBI-HH-M-291/2000, Department of Computer Science, University of Hamburg, Germany, 2000.
25. T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
26. R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on System, Man, and Cybernetics*, 19(1):17–30, 1989.
27. P. Resnik. Using information content to evaluate semantic similarity in a taxanomy. In *Proc. of the Int. Joint Conf. for Artificial Intelligence (IJCAI-95)*, pages 448–453, 1995.
28. P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.
29. A. Tversky. Features of similarity. *Psycological Review*, 84(4):327–352, 1977.

A Appendix Foundation

Description Logics are a family of logic languages with different expressive power, depending from the constructors that are allowed for building complex concept descriptions. We will mainly focus on $\mathcal{ALE}$ and $\mathcal{ALC}$ logic. Even if they are sub-languages of OWL[12], they are considered a good compromise between expressive power and computational complexity required by the inference operators [12].

In DLs, descriptions are inductively defined starting with a set N_C of *primitive concept* names and a set N_R of *primitive roles*. The semantics of the descriptions is defined by an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set representing the *domain* of the interpretation, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps each $A \in N_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each $R \in N_R$ to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The *top* concept $\top$ is interpreted as the whole domain $\Delta^{\mathcal{I}}$. The *bottom* concept $\bot$ corresponds to $\emptyset$. Complex descriptions can be built in $\mathcal{ALC}$ using primitive concepts and roles and the following constructors whose semantics is also specified. The language supports *full negation*, denoted $\neg C$ (given any description C), it amounts to $\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$; concept *conjunction*, denoted $C_1 \sqcap C_2$, yields an extension $C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$; concept *disjunction*, denoted $C_1 \sqcup C_2$, yields the union $C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$; *existential restriction*, denoted $\exists R.C$, is interpreted as the set $\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$ and the *value restriction* $\forall R.C$ that has the extension $\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$.

$\mathcal{ALE}$ logic is a sub-language of $\mathcal{ALC}$ as only a subset of $\mathcal{ALC}$ constructors is allowed. Specifically, concept disjunction is not allowed and only the *atomic negation* can be used, namely complex concept descriptions cannot be negated.

The main inference in DLs is *subsumption* between concepts:

Definition 9 (subsumption). *Given two descriptions C and D, C* subsumes *D, denoted by $C \sqsupseteq D$, iff for every interpretation $\mathcal{I}$ it holds that $C^{\mathcal{I}} \supseteq D^{\mathcal{I}}$. When $C \sqsupseteq D$ and $D \sqsupseteq C$ then they are equivalent, denoted with $C \equiv D$.*

A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is the set of definitions $C \equiv D$, meaning $C^{\mathcal{I}} = D^{\mathcal{I}}$, where C is the concept name and D is its description. $\mathcal{A}$ contains assertions on the world state, e.g. $C(a)$ and $R(a,b)$, meaning that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. Subsumption based axioms ($D \sqsubseteq C$) are also allowed in the TBoxes as partial definitions.

Another inference operator is the *instance checking*, that is deciding whether an individual is an instance of a concept or not [12]. Conversely, the *Most Specific Concept* is the most specific description (w.r.t. the subsumption relationship) of which an individual is instance of.

Definition 10 (Most Specific Concept). *Given an ABox $\mathcal{A}$ and an individual a, the most specific concept of a w.r.t. $\mathcal{A}$ is the concept C, denoted $MSC_{\mathcal{A}}(a)$, such that $\mathcal{A} \models C(a)$ and $\forall D$ such that $\mathcal{A} \models D(a)$, it holds: $C \sqsubseteq D$.*

In the general case of a cyclic ABox expressed in a DL endowed with existential or numeric restriction, the MSC cannot be expressed as a finite concept description [12], it can only be approximated [24]. The *Least Common Subsumer* is the most specific concept subsuming all concept descriptions in given set:

Definition 11 (Least Common Subsumer). *Let $\mathcal{L}$ be a description logic. A concept description E of $\mathcal{L}$ is the **least common subsumer (LCS)** of the concept descriptions $C_1, \cdots, C_n$ in $\mathcal{L}$ ($LCS(C_1, \cdots, C_n)$ for short) iff it satisfies:*

1. *$C_i \sqsubseteq E$ for all $i = 1, \cdots, n$ and*

[12] For OWL we mean OWL-DL that is the one allowing enough expressive power without loosing decidability of reasoning procedures.

2. *E is the least $\mathcal{L}$-concept description satisfying (1), i.e. if E' is an $\mathcal{L}$-concept description satisfying $C_i \sqsubseteq E'$ for all $i = 1, \cdots, n$, then $E \sqsubseteq E'$.*

Depending on the DL language, the LCS needs not always exist. If it exists, it is unique up to equivalence. In $\mathcal{ALC}$ and $\mathcal{ALE}$ logic, the LCS always exists [1, 12]. In $\mathcal{ALC}$ (as in every DL allowing for concept disjunction) the LCS is given by the disjunction of the considered concepts. In $\mathcal{ALE}$, where disjunction is disallowed, the LCS is computed by taking the common concept names in the concept descriptions (also in the concepts scope of universal and existential restrictions w.r.t. the same role), without considering the TBox (see [1] for more details). The $\mathcal{ALE}$ LCS computed using such an approach often results to be very general. For this reason the notion of LCS computed w.r.t. the TBox[13] has been introduced [2].

Definition 12 (LCS w.r.t. a TBox). *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be DLs s.t. $\mathcal{L}_1$ is a sub-DL of $\mathcal{L}_2$. For a given $\mathcal{L}_2$-TBox $\mathcal{T}$, let $\mathcal{L}_1(T)$-*concept descriptions *be those $\mathcal{L}_1$-concept descriptions that may contain concepts defined in $\mathcal{T}$. Given an $\mathcal{L}_2$-TBox $\mathcal{T}$ and $\mathcal{L}_1(T)$-concept descriptions $C_1, \ldots, C_n$, the least common subsumer (LCS) of $C_1, \ldots, C_n$ in $\mathcal{L}_1(T)$ w.r.t. $\mathcal{T}$ is the most specific $\mathcal{L}_1(T)$-concept description that subsumes $C_1, \ldots, C_n$ w.r.t. $\mathcal{T}$, i.e., it is an $\mathcal{L}_1(T)$-concept description D such that:*

1. *$C_i \sqsubseteq_{\mathcal{T}} D$ for $i = 1, \ldots, n$*
2. *if E is an $\mathcal{L}_1(T)$-concept satisfying $C_i \sqsubseteq_{\mathcal{T}} E$ for $i = 1, \ldots, n$, then $D \sqsubseteq_{\mathcal{T}} E$*

In [2], the case $\mathcal{L}_2 = \mathcal{ALC}$ and $\mathcal{L}_1 = \mathcal{ALE}$ is focused and a brute force algorithm for computing the LCS is defined which is hardly practically usable. For this reason, an algorithm for computing an approximation of the $\mathcal{ALE}$ LCS w.r.t. an $\mathcal{ALC}$ TBox has been presented. Such an approximation is called *Good Common Subsumer* (GCS) w.r.t. a TBox [2]. It is computed by determining the smallest conjunction of (negated) concept names subsuming the conjunction of the top level concept names of each considered concept, and the same for the concepts that are range of role restrictions w.r.t. the same role. The GCS is more specific than the LCS computed by ignoring the TBox.

[13] The TBox can be described by a DL that is more expressive than $\mathcal{ALE}$.